3

The BBN TransTalk Speech-to-Speech Translation System

David Stallard et al.,*
Raytheon BBN Technologies, USA

1. Introduction

Portable translation devices which enable people who speak different languages to communicate with each another in voice are likely to have a far-reaching impact in both the civilian and military worlds. While long a staple of science fiction, a la the "Star Trek universal translator", devices that actually translate between languages have not been fully realized. In the last decade, however, under the sponsorship of the Babylon and TRANSTAC (Translation for Tactical use) programs of the US Defense Advance Research Program Agency (DARPA), several research sites, including BBN (Stallard et al., 2007), CMU (Waibel et al., 2003), IBM (Gao et al., 2006), SRI (Akbacak et al., 2009) and USC (Belvin et al. 2005), have made significant progress in developing a real two-way speech-to-speech (S2S) translation systems. These systems are not "universal translators" in the science-fiction sense, in that must be configured for the language and conversational domain of interest, rather than spontaneously understanding them. However, the technology is language-independent, and under the auspices of the TRANSTAC program, systems have been configured for several different foreign languages of interest to the US Government, including Iraqi Arabic, Malay, Farsi, Dari, and Pashto. Though the technology is also domain-independent, most of these systems support conversations in the so-called "force protection" military domain, which is broadly construed to include not only conversations relevant to checkpoints, searches, and other military operations, but also rapport building, civil affairs, and basic medical conversations.

In this article, we describe BBN's S2S system, TransTalk, which runs not only on laptops and ultra-mobile PCs, but also on mobile Android Smartphones, running locally on the device itself and not a server. In common with other TRANTSTAC systems, TransTalk's technology is language-independent, and has been configured to translate between English and numerous other languages, including Iraqi Arabic, Dari, Pashto, Farsi, and Malay. TransTalk also has a uniquely simple user interface which does not require the user to view a screen. TransTalk integrates automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) synthesis engines for converting speech in one language into a different language. In particular, TransTalk uses the BBN Byblos ASR engine for converting speech to text. BBN Byblos is a multi-pass, speaker-independent large

*Rohit Prasad, Prem Natarajan, Fred Choi, Shirin Saleem, Ralf Meermeier, Kriste Krstovski, Shankar Ananthakrishnan and Jacob Devlin

vocabulary speech recognizer which uses n-gram language models instead of a finite state grammar. For machine translation, TransTalk primarily uses a BBN-developed Statistical Machine Translation (SMT) component. For text-to-speech synthesis, we use engines developed by TTS research sites under the DARPA TRANSTAC program, which includes CMU and Cepstral. TransTalk has consistently been a top performer in independent applications conducted by the US government.

Of particular importance to the recent progress in S2S technology has been the adoption of Statistical Machine Translation (SMT). SMT uses a statistical, corpus-driven approach, rather than hand-coded translation rules; and is driven by automated rather than manual performance evaluation. There are two advantages conferred by SMT. First, the statistical paradigm generally provides better performance than approaches based on hand-written rules, as these rules are often brittle and conflict with one another. Second, the automated nature of the process allows much more rapid development and testing of new approaches to improve performance. The resulting labor savings has greatly accelerated the progress of both machine translation and S2S as a whole. In this way, SMT may be seen as following in the footsteps of ASR, which also underwent dramatic improvement following the adoption of statistical paradigms and automated evaluation.

S2S systems are configured for particular language pairs by training ASR and translation models using speech and language data (recordings, transcriptions, and translations) in the relevant languages. For optimal performance, the data collected should match the intended domains of conversation of the system. That is, if the intended domain is force protection, data should be collected for that domain. While data outside that domain can be helpful for general modeling of the given language, it often lacks the key concepts and constructions that are important in the specific application domain. In practice therefore, the domain-relevant data is frequently collected in simulated translingual dialogs between role-playing individuals. Because of the finite resources available for such data collection, and because many of the languages of interest are low-resource themselves, S2S technology must frequently cope with sparse data, which poses challenges for both ASR and MT.

In this paper, we describe the individual components of our system, including its ASR, MT, TTS, dialog manager, and user interface components, and their integration into a free-form two-way S2S translation system. We present novel algorithms for overcoming specific challenges posed by colloquial and low resource nature of the languages of interest. Another important issue in speech-to-speech translation is using some form of confirmation strategy for minimizing errors in transferring a concept from one direction to other. Such errors can easily cause the dialog to drift or stall. Here, we present multiple confirmation techniques for a user to get feedback from the system so as to detect errors in the concepts being conveyed by the system. In addition, we describe a novel methodology for assessing the usefulness of these user confirmation strategies.

The remainder of the paper is organized as follows. Section 2 gives a brief overview of the our TransTalk system. Section 3 discusses our user-centered approach to system design and interaction. Section 4 presents work we have done on our ASR component, with particular emphasis on improving performance for colloquial dialects of low-resource languages. Section 5 discusses the machine translation component of our system, and continues the emphasis on low-resource languages. Section 6 presents live evaluation results for our system, and Section 7 concludes.

2. System overview

A block diagram of the BBN TransTalk system is shown in Figure 1. The BBN TransTalk system uses BBN's Byblos speech recognizer (Nguyen and Schwartz, 1997), BBN's SMT engine, and third-party text-to-speech synthesizer(s). Various input modalities are supported, including both handheld and headset microphones. The primary physical interface is the "BBN SuperMic", a handheld unit developed by BBN, which encompasses a directional microphone, speakers, and two push-and-hold "listen" buttons, one for receiving the English speech, and the other for receiving the foreign speech. Figure 2 shows the BBN TransTalk system running on multiple platforms: (a) Ultra-Mobile PC (UMPC) with BBN SuperMic, and (b) Android smartphone.

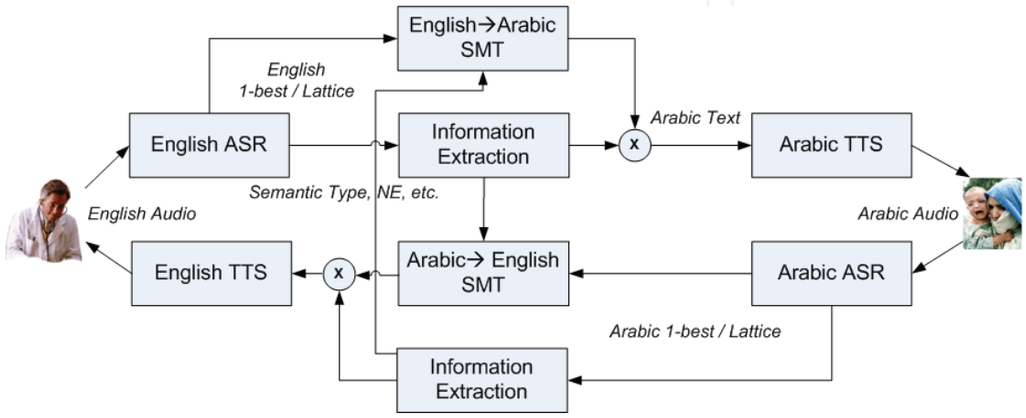

Fig. 1. Block Diagram for BBN TransTalk 2-Way S2S Translator.

English speech received through the physical interface device is sent to the English speech recognizer, which outputs a sequence of words in the recognizer's vocabulary. This English text is then sent to both the SMT component and to a separate information extraction component. The information extraction component performs the following functions:

1. English speech received through the physical interface device is sent to the English speech recognizer, which outputs a sequence of words in the recognizer's vocabulary. This English text is then sent to both the SMT component and to a separate information extraction component. The information extraction component performs the following functions:
2. Canonicalization: matches the utterance to one of a set of utterances for which it has stored translations (Stallard et al., 2007). If none is found, the output of the SMT component is used instead. Arabic speech corresponding to the translation is then played back. This may be a pre-recorded wave file, or more generally, the result of text-to-speech synthesis.
3. Question Detection: determines whether the recognized utterance is a question or a statement. This module also classifies the question as one of the pre-defined classes.}
4. Named Entity Detection: detects whether the spoken response has named entities such as person names, place names, geo-political organizations, etc. (Prasad et al., 2008; Bikel et al., 1999).

A composite foreign language translation ("Arabic") in Figure 1 is produced by the SMT and the information extraction component. This translation is then either played as a pre-recorded wave file, or more generally, by the text-to-speech synthesis.

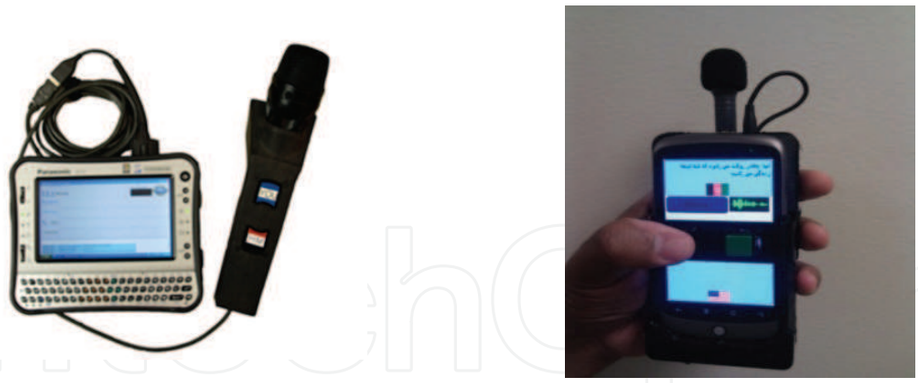

Fig. 2. BBN TransTalk Two-way S2S Translator on UMPC and Smartphone platform.

The foreign language speaker's reply ("Arabic" in Figure 1) is sent to the foreign language recognizer, which outputs text in the foreign language. The foreign language text is translated into English text with a process similar to the one in English-to-Foreign direction. The translated text is then sent to a second speech synthesizer, which speaks it out for the English speaker to hear, and/or displaying it on a screen.

3. User-centered interaction

3.1 Overall design

A key aspect of our TransTalk system is its user-centered interface. Our design of the interface was guided by a number of desiderata. Obviously, the interface had to be simple, easy to use, and efficient. But it also had to work without a screen display, so that it could be used by soldiers in "eyes-free" operation. We also wanted it to be easy for the user to detect when the system had made an error in translation, and to correct the error. Finally, we wanted the user to be able to abort system output and barge in at any time to speak again, no matter what the system was doing at the time. The user interface design we developed provides an elegant joint solution to all of these goals.

In physical terms, the system's user interface is indeed simple. It consists of just two push-and-hold buttons, one labeled "YOU", the other "HIM". The YOU button commands the system to begin listening for English speech, and performing ASR on it as soon as speech is heard. The HIM button, similarly, causes the system to begin listening for speech in the foreign language and performing ASR on it. When the button is released, the system stops listening, finishes up ASR, and then passes the ASR output to MT for translation into the opposite language, and then to TTS for speaking out to the other party.

The requirement that the ELS be able to detect translation errors has impact on the interface's behavior. It is simply a fact that despite ongoing improvements in the underlying technologies, for the foreseeable future, S2S systems will make errors in translation. If undetected, translation errors can lead to mutual incomprehension and a complete breakdown of the dialog. To cope with this problem, our system presents the ELS with a "confirmation" utterance that tells him what the system thought he said, so that he can determine whether the system made an error. In the system's usual mode of operation, this confirmation utterance is simply a read-back of the English ASR result. (We discuss alternatives such as "back-translation" in a later section).

Now, if a display screen were available, this confirmation utterance would simply be displayed on the screen and the ELS could quickly scan it to determine whether he had been

understood correctly. However, the TRANSTAC program requires (and most military users prefer) that the system be usable without a display. The only way the confirmation utterance can be delivered is through voice. One possibility would be for the system to explicitly ask the user via TTS "Did you say 'Show me your id'?". However, this "explicit confirmation" would slow down the dialog, and the repeated confirmation interactions would likely be irritating to the user. Our system instead uses "implicit confirmation", in which it speaks out the confirmation utterance for the ELS to hear. If the ELS decides this is correct, he takes no action, and the system's processing continues as normally, generating the translation and playing it out for the foreign language speaker (FLS) to hear. If the ELS instead decides that the confirmation is incorrect, he simply presses the "YOU" button again. This aborts any ASR, MT, or TTS activities the system may be performing, and in particular halts voice output in either English or the foreign language. The system then begins listening to the ELS, who may speak again, either repeating his utterance more clearly, or rephrasing it, as he chooses.

Because it aborts all ongoing system activities, the "YOU" button effectively doubles as an "abort" button. If the ELS wants to abort the system's current activities, but does not want to speak again right away, he can simply press the YOU button and then quickly release it again, without speaking. The system recognizes this very short listening interval as being an abort, rather than a speech event, and ignores the empty result that ASR returns. In this way, we avoid the need for a dedicated third "abort" button, thereby retaining our maximally simple two-button interface.

The above-described abort and barge-in functionality of the YOU button illustrates another key design goal of our system, which might be stated as "The user controls the system; the system does not control the user". That is, the system does not constrain the ELS user's actions, but rather allows him to interrupt it at any time, and speak again without having to wait for the system to be "ready". He can simply assume that the system is always ready.

Such a capability is not straightforward to achieve, however. In the synchronous pipeline of ASR, MT, and TTS, the various components can be in different states when the abort/barge-in occurs. Example states include processing the last input, returning results for the last input, aborting in response to the button push, and resetting internal data structures to prepare for the next input. A component that is in any of these states will not be ready to begin immediately processing new input. The easiest way to cope with this might be to require all system components to return to their ready state before allowing new input from the user (perhaps using a beep as a ready signal). However, the time that it would take all the system's components to return to their ready state is variable, and depends among other things upon which component(s) were interrupted, and which state they were in when interrupted. To force the ELS to wait until all components are in their ready state before he speaks again, even by 10's of milliseconds, is to invite user frustration and user error. In particular, it would be very difficult to prevent the user from speaking too early, thus resulting in truncated utterances being sent to the ASR, with the consequent loss of speech recognition and translation accuracy.

Our approach avoids these problems. Instead of attempting to configure the user's behavior to cope with the situation, we configure the system's internal behavior. That is, if a component is not yet ready for new input, the system buffers the input until the component returns to the ready state and can begin processing it. Examples include user speech (for ASR), English ASR output (for MT), and foreign-text MT output (for TTS).The system begins to buffer user speech, in particular, as soon as the button is pressed. Any slight latency that may results from this

internal wait will be manifested only as a slight delay in the system's final output, which will probably not be noticeable by the user, rather than a delay enforced on the user's input, which would certainly be noticeable by him. The overall theme of our interface is that the system retains its internal complexity inside itself, where it belongs, rather than imposing it upon the user, who has more important things to do.

3.2 Improving the efficiency of voice confirmation

An obvious efficiency issue presented by voice confirmation is the additional time it costs the interaction. Confirming the ELS's utterance means that each English utterance is spoken twice, first by the ELS, and then again by the system. To alleviate this, we can substantially reduce the effective time that confirmation costs by performing the E2F MT concurrently with the confirmation TTS. Since the E2F MT would have to be performed anyway whether or not confirmation was done, doing it at the same time as confirmation, rather than waiting until confirmation is done, saves time. Effectively, we are reducing the time that confirmation TTS costs the dialog (avg. 3.0 seconds), by subtracting the time that the MT takes to run (avg.1.2 seconds), yielding a relative reduction of 40%. Figure 3 illustrates this time savings, with the top panel representing confirmation and MT running in series, compared with confirmation and MT running in parallel.

Moreover, if the ELS and FLS are listening on separate channels, as is the case in the two-phone configuration, we can obtain even greater time savings by also playing the Iraqi translation TTS itself at the same time as the English confirmation TTS. In this mode, the system begins playing the foreign TTS as soon as the E2F MT produces the foreign-language utterance for it. Given that E2F is faster than confirmation, it will generally be the case that the system will then be speaking English and foreign-language utterances simultaneously. However, because each party has his own phone, neither hears the TTS output intended for the other. Because the system is speaking in two different languages in parallel, we term this technique "parallel confirmation".

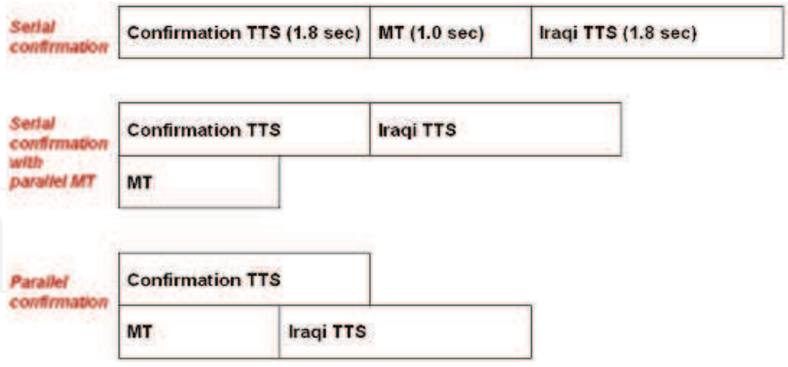

Fig. 3. Different Confirmation Modes.

Figure 3 illustrates the benefits of this technique by graphically comparing Serial Confirmation, Serial Confirmation with Parallel MT, and Parallel Confirmation. Note that Parallel Confirmation effectively reduces the time-cost of confirmation to zero, since in this configuration the confirmation is performed entirely in parallel with activities that would need to be performed anyway, and whose combined duration exceeds that of the confirmation.

Note also an additional important benefit of parallel confirmation; namely, that it enables improvements in the MT's speed to increase the system's overall translation speed,

specifically, by reducing the lag before foreign TTS starts. By contrast, in Serial Confirmation with Parallel MT, no further throughput increase is possible, once the time that the MT takes to translate the ELS's English utterance is less than the time that the English confirmation TTS takes to speak that utterance.

We have used the parallel confirmation effectively in a two-phone configuration, where each speaker has his own handset. The two phones communicate via Bluetooth and send text back and forth, allowing confirmation TTS and translation TTS to be generated in parallel on the respective handset.

3.3 Generating the confirmation utterance

As stated previously, our primary method of generating a confirmation utterance is to simply read back the ASR output. The advantage of this approach is that it allows the user to catch ASR errors quickly, which is important since ASR errors on concept words guarantee a wrong translation. The disadvantage, of course, is that correctness of the ASR output says nothing about whether the MT output itself was correct. In particular, it is perfectly possible for the ASR output to be error-free, yet for the MT result for that output to be completely wrong.

An alternative strategy that addresses this issue is "back-translation", in which the output of ASR + MT into the target language is translated back into the source language, and the result played back to the user for his approval. A back-translation that is close to the original utterance in meaning can have the important psychological advantage of making the user feel more secure that he was translated correctly. In fact, in informal interviews, users of our S2S system who have use system with the back-translation alternative to cofirmation express a strong approval of it.

The back-translation approach can be objected to, however, on two grounds.The first objection is that the output of back-translation, having been passed through three successive noisy channels (ASR, forward MT, and backward MT), will likely be hopelessly garbled, causing most forward translations to appear wrong. The second objection is that even when not garbled, back-translation may yet be misleading, since the same incorrect phrase pair rule used in the forward direction may also be selected (in reversed form) in the backward direction, leading back to the original source phrase again and leaving the error undetected.

The most straightforward way to evaluate the efficacy of back-translation and other confirmation approaches would be to run two complete sets of live evaluations, one with the approach in question and one without, and compare the results on measures such as concept transfer, rate of concept transfer, user satisfaction, and the like. Unfortunately, given the great expense of carrying out realistic evaluations, which involve assembling personnel from multiple locations in the US, this is infeasible. We therefore must look for some offline method of evaluating back-translation.

The implicit hope of back-translation, and indeed of any other confirmation utterance-based strategy, is that the ELS could develop a mental model, based on his experience in using the system, of the minimal level of back-translation quality that would predict that the forward translation is correct. We do not concern ourselves here with how the ELS would develop such a mental model, but rather seek to determine whether it is even possible to develop such a model at all – that, to determine whether the data is sufficiently consistent that is possible to infer useful prediction rules from it.

As a subjective measure of translation quality in either direction, we use the familiar 1-5 Likert scale to rank both forward translations (i.e. English-to-foreign), and back-translations. We do not assume that users will actually assign Likert scores while using the system; but

instead view the score as a numerical proxy for the user's reaction. We assign the following interpretations to the different elements of the Likert scale.

5: Essentially a perfect translation.
4: An adequate if slightly disfluent translation which conveys the utterance's meaning
3: A partial translation which is missing one or more concepts, or is severely disfluent.
2: A translation which is missing most of the concepts.
1: A translation with no apparent relation to the input.

If back-translation were a perfectly effective diagnostic, the Likert rating of the back-translation and the Likert rating of the forward translation would have the same value. Obviously, this will seldom be the case, since both are noisy processes, with one of them operating on the output of the other. One might then fall back to a weaker requirement, for example only requiring that the back-translation and forward-translation quality be well-correlated in a linear relationship.

Our approach to this problem is quite different. Note that we are not interested in predicting the actual value of the Likert rating for the forward translation, but rather in simply predicting whether or not the forward translation's Likert rating is above a certain threshold of acceptability. Therefore, we seek to use the back-translation for binary classification, rather than regression.In particular, we choose a specific minimum acceptable Likert score *F* for the forward translation - say, a score of 4. We then test various minimum thresholds *B* for the back-translation Likert score. In particular, for utterances whose back-translation score is at or above the threshold *B*, we test the prediction that the utterance's forward translation Likert score will be at or above the threshold *F*, and thus acceptable. Below *B*, we predict that the forward translation Likert will be below *F*, and therefore unacceptable. We compute precision, recall, and F-measure for each such threshold. Different costs for the different error types, e.g. a higher penalty for false acceptance than for false rejection, can be straightforwardly taken into account by using a weighted harmonic mean.

To test the evaluation methodology outlined above, we used a set of 779 English utterances that were spoken to our system by ELS users during the TransTac live evaluation in June 2008, conducted by the US government's National Institute of Standards and Technology (NIST). In this evaluation, active-duty military personnel played the part of the ELS, while native speakers of Iraqi Arabic were recruited to play the part of the FLS. The utterances of both parties, and the system's ASR and MT outputs for these, were recorded for later analysis. The Iraqi translation output produced by the system for the ELS's utterances was Likert-scored by a native Arabic speaker experienced in the application domain. To produce the back-translations, we ran (offline) our Arabic-to-English MT on the system's Iraqi translation outputs. These back-translations were then Likert-scored by a native English speaker knowledgeable in the application domain. For comparison, the same English ranker also Likert-scored the output of our system's English ASR for these same 779 utterances (i.e. the ASR read-back strategy).

The resulting scores are shown in Table 1. As can be expected, the highest mean Likert scores were produced on ASR output, which tends to overestimate the true (forward) Likert score, while the lowest were associated with back-translation output, which tends to underestimate it. Both were approximately equally well-correlated with forward Likert score, however, with a correlation coefficient of approximately 0.60. The English ASR WER obtained on this corpus was 6.2%, while the English-to-Arabic BLEU score on this ASR output was 56.7%.

Some examples of back-translations and their Likert rankings are: "Turn off your vehicle" (for "Turn your vehicle off"), ranked 5; "Construction prior experience do you have" (for "Do you have prior construction experience"), ranked 4; and "How many subcontracting work" (for "How many subcontractors work for you") ranked 3. Table 1 shows the mean Likert scores for each of the conditions, namely, forward translation, back-translation, and ASR output of Likert scores for the back-translation.

To obtain results on back-translation efficacy, we set the forward translation Likert score threshold F to be 4.0. This may be considered a good minimum acceptable score for our purposes, as scores below 4.0 are by definition associated with "semantic damage" to the translation. Table 2 gives acceptance rate, false rejection rate, false acceptance, F-measure, and precision-weighted F-measure for different back-translation Likert score cutoffs *B*. Each row of this table can be interpreted as a prediction rule, which predicts that an utterance whose back-translation Likert score is at or above the cutoff will have a forward translation whose Likert score will be 4.0 or higher.

Forward-trans.	Back-trans.	ASR
4.42	3.99	4.64

Table 1. Mean Likert Scores.

For many S2S applications, a false acceptance can be regarded as worse than a false rejection, because of the possibility of confusing the respondent, etc. For example, one might decide that a false acceptance is twice as bad as a false rejection. The rightmost column of Table 2 gives F-measure computed with these weights (0.67 vs. 0.33).

The results in Tables 1 and 2 seem to show that the worst fears regarding back-translation are not realized. By no means does back-translation yield incomprehensible utterances for most cases, nor is it a false and over-optimistic guide. Indeed, for cutoffs of 4.0 or higher, its false acceptance rate is actually quite low. This precision does come at the expense of recall, however, and in particular at a cutoff of 4.0 fully 39% of ELS utterances would be rejected and have to be retried. A better strategy might be a slightly less strict cutoff of 3.5, which yields a low false acceptance rate of 8%, while falsely rejecting only 14%. This rule corresponds to a back-translation which subjectively seems rather poor, but which is not completely deficient.

Cutoff	Acpt	FlsRej	FlsAcc	FMsr	WFMsr
5.0	0.27	0.68	0.02	0.48	0.58
4.5	0.35	0.59	0.03	0.57	0.67
4.0	0.61	0.29	0.04	0.81	0.86
3.5	0.78	0.14	0.08	0.89	0.90
3.0	0.94	0.02	0.14	0.92	0.90
2.5	0.96	0.02	0.15	0.91	0.89
2.0	0.99	0.00	0.16	0.91	0.88
1.0	1.00	0.00	0.17	0.91	0.88

Table 2. Precision and Recall for Back-translation.

The key question to be addressed, however, is whether back-translation is better than our default confirmation strategy of simply reading back the system's English ASR output. To

address this question, Table 3 repeats the above experiment, using Likert rankings on the system's English ASR output. Note the false acceptance rate is higher than for back-translation, but the false rejection rate is much lower, yielding good F-measure scores at all values of the cutoff. For most cutoff values, the ASR read-back strategy even slightly out-performs back-translation on weighted F-measure.

Cutoff	Acpt	FlsRej	FlsAcc	FMsr	WFMsr
5.0	0.72	0.19	0.07	0.86	0.88
4.5	0.75	0.17	0.08	0.88	0.89
4.0	0.86	0.07	0.10	0.92	0.91
3.5	0.94	0.02	0.13	0.92	0.90
3.0	0.99	0.00	0.16	0.91	0.89
2.5	0.99	0.00	0.16	0.91	0.89
1.0	1.00	0.00	0.17	0.91	0.89

Table 3. Precision and Recall for ASR Readback.

It might seem from this analysis that ASR read-back is a superior strategy to back-translation. It should be noted, however, that ASR read-back on this dataset has a floor of 7% false acceptance, below which it cannot possibly go. The back-translation strategy, by contrast, can go as low as 2% false acceptance, albeit at the price of a very high false rejection rate. If the goal were to fix a certain maximum allowable rate of false acceptance rate – say 8% - rather than maximizing F-measure, the back-translation strategy could be seen as slightly superior, resulting in a 14% false rejection rate as opposed to ASR read-back's 17%.

4. Automated Speech Recognition (ASR)

4.1 Overview

The ASR component of our S2S system is the BBN Byblos speech recognizer (Nguyen and Schwartz, 1997). Byblos models speech as the output of context-dependent phonetic Hidden Markov Models (HMMs). The outputs of the HMM states are mixtures of multi-dimensional diagonal Gaussians. Different forms of parameter tying are used in Byblos, including State Tied Mixture (STM) triphone and State Clustered Tied Mixture (SCTM) quinphone models. The mixture weights in both these cases are shared based on decision tree clustering using linguistic rules. Decoding is performed using our patented two pass search strategy (Nguyen and Schwartz, 1997). The forward pass is a fast-match beam search using an STM acoustic model and an approximate bigram language model. The output of the forward pass consists of the most likely word-ends per frame along with their partial forward likelihood scores. The backward pass operates on the set of choices from the forward pass to restrict the search space, and uses the more detailed SCTM quinphone model and a trigram language model to produce the best hypothesis.

Development of ASR capability for our S2S system posed special challenges, as many of the languages of interest, including Iraqi Arabic, Pashto, and Dari ,are not only low-resource, but also of colloquial dialect. Such languages are challenging for ASR development for two reasons. First, in many cases there is no standard written form for the colloquial dialects of a language, leading to a lack of consistency in transcriptions of audio in that language. Second, creating pronunciation lexicons for words in these dialects is challenging. Most ASR engines

use phones as units for acoustic modeling, and each word in the recognition lexicon is manually spelled using these phones. Given that skilled acoustic-phoneticians for low-resource languages are few, the manual creation of phonetic spellings for large vocabulary ASR in low-resource languages is generally impractical. Moreover, creating a phonetic dictionary is even more difficult for languages that use the Arabic script for their writing system. This is because in most of the colloquial dialects of such languages (e.g. Iraqi Arabic, Farsi, Dari, and Pashto), short vowels do not correspond to characters of their own, but instead appear as diacritic marks on other characters, and furthermore, are usually omitted. This results in additional pronunciation ambiguity and language-model confusability for vowel-less word forms which may correspond to several different actual words. A a classic example is the Arabic root form having the meaning of writing or inscribing, "k-t-b", which can appear with many different vowel forms, some of which correspond to the "book", "writer", "he wrote", "bookdealer", etc. Nevertheless, all these forms are typically written simply as "ktb".

Given these challenges, most state-of-the-art ASR systems resort to the "grapheme-as-phoneme" approach for lexicon creation (Billa et al., 2002).In this approach, the pronunciation for a word is derived directly from the orthography by treating the constituent character/grapheme as phones. The grapheme approach has several advantages including: (1) it automates the dictionary creation process, thereby simplifying the ASR training, (2) it does not suffer from inter-annotator differences in manual pronunciation creation for words, and (3) it allows the automated addition of new vocabulary at runtime.

While the grapheme-as-phoneme approach has emerged as a promising approach for mitigating the impact of inherent ambiguity introduced by absence of short vowels, researchers have also explored automatic diacritization based on morphological analysis (Xiang et al., 2006). However, such automatic diacritization methods have several shortcomings, - most of which are due to the creation of large number of vowelization variants, of which very few are actually useful. The increased number of pronunciation variants for a given word has several undesirable effects. First, it typically increases the word's confusability with other words, because the difference in pronunciation between words usually becomes smaller. Second, it increases the search space during decoding, because the decoder has to consider a larger number of pronunciations for each word. Finally, most of the rules used by morphological analyzers for a given language were developed for the language's formal form, and tend to break down when applied to colloquial dialects. Therefore, the grapheme approach is usually still better than using automatic diacritization for colloquial dialects. Nevertheless, the recognition performance with grapheme-as-phonemes is significantly worse than with a high-quality, manually created phonetic dictionary.

In this section, we present techniques that reduce the difference between grapheme and full phonetic systems by using manual pronunciations for only a small fraction of words (Prasad et al, 2010). Specifically, we investigate two different techniques for developing a recognizer for colloquial Pashto. The first technique uses a modified version of the text-to-phoneme (T2P) tool (Black et al., 1998). T2P is a decision tree approach that learns letter-to-sound rules from a small set of manual pronunciations. The standard version of T2P has serious limitations for languages for which the number of letters/graphemes is significantly different from the number of phones. Here, we describe a novel approach for extending T2P to deal with such languages. The second technique uses a hybrid phoneme/grapheme recognition approach, similar to the one described in (Magimai-Doss et al., 2004).

4.2 Automated lexicon creation

Grapheme-as-Phoneme: We developed a grapheme-as-phoneme (Billa et al., 2002) mapping based on the orthography of the words in the Pashto data in the TRANSTAC corpus. The Pashto corpus spans a wide range of scenarios, including checkpoint patrols, civil affairs, medical interviews, facility inspections, etc. The audio in the corpus was segmented and transcribed by Appen, Pty, Ltd. We first pre-processed Appen's audio data and transcriptions in order to eliminate segments with transcriptions that are not suitable for either acoustic or language model training, e.g. unintelligible speech, long pauses, overlapping, or foreign speech. Next, we divided the speakers and data into two sets: A 34-hour (400K total, 10K unique words) training set and a 2-hour, 26K total word, test set.

We used a modified Buckwalter transliteration system to create Romanized forms of Pashto letters. A total of 34 phones were derived from the graphemes after Romanization. Because several letters map to the same sounds, the total number of graphemes is less than the total number of letters in Pashto alphabet. In Table 4, we compare the phone set used for the phonetic and grapheme representations.

Representation	Pashto Sounds	Non-speech	Total Phonemes
Phonetic	42	3	45
Grapheme	34	3	37

Table 4. Pashto phoneme and grapheme representation

Learning Text-to-Phoneme Mappings: Our approach for text-to-phoneme conversion is based on the set of public-domain tools from CMU (Black et al, 1998). The training of T2P models with the CMU tools is performed in three steps:

1. Align letters to phonemes in the training dictionary
2. Extract contextual features from the alignments
3. Train a decision tree using the contextual features.

We found serious limitations in the alignment step of the standard T2P tool. Specifically, the standard alignment process can only handle word and pronunciation pairs where the number of letters is greater or equal to the number of phones, allowing no more than one phone to be aligned to a given letter. While this may be acceptable for most of English words, it does not work for many other languages including Pashto.

Therefore, we implemented a new alignment algorithm that overcomes the limitations of the standard T2P tool. The algorithm uses iterative expectation maximization (EM) style optimization to find alignments that best describe the training dictionary. Our updated alignment algorithm has the following key steps:

1. Initialization
   a. Set P(phone = p | letter = c) = $\frac{num\ dictionary\ pairs\ with\ both\ c\ and\ p}{num\ words\ with\ c}$
   b. Set P(deletion | c) = 0.1
2. Iterate until convergence
   a. Find best path (according to the current model) through [letters phones] grid using dynamic programming and allowing any number of phones per letter
   b. Update P(p | c) = $\frac{number\ of\ aligned\ pairs\ (p,c)}{total\ number\ of\ c}$
   c. Update p(deletion | c) = $\frac{number\ of\ unaligned\ c}{total\ number\ of\ c}$

Hybrid Phoneme/Grapheme: In the hybrid phoneme/grapheme approach, during recognition each word is modeled with two different phone sequences. The first phone sequence is created

manually by native speakers. The second sequence uses a grapheme-as-phone representation. In training, we assume independence between the manual phone set, "*P*" and the grapheme representation, "*G*", and train two different sets of context-dependent HMMs. Words which do not have any manually created pronunciations are spelt with just the grapheme-derived phones. While performing recognition, one can also use pronunciation probabilities to weight the grapheme and phoneme pronunciations differently.

4.2 Experimental results

In the following, we present experimental results on Pashto ASR for comparing the different approaches outlined above. All recognition experiments used a three pass recognition strategy in the BBN Byblos recognizer. The first pass, referred to as the forward pass, uses context-dependent triphones with state-tied mixture (STM) parameter tying and a bigram language model (LM). The second pass, referred to as the backward pass, operates on the lattice from the forward pass using context-dependent quinphones with SCTM configuration for acoustic models and a trigram LM. The output from the backward pass is a lattice or an n-best list. The third and final recognition pass, referred to as the rescoring pass, uses SCTM models trained with crossword quinphones to re-rank the n-best list produced by the backward pass. All acoustic models in the results below were trained using maximum likelihood estimation (MLE). LM training used a total of 700K words from Pashto transcriptions and translations available in the corpus provided by Appen.

Our first experiment was designed to compare the quality of pronunciations produced by standard T2P and the modified version using the improved alignment algorithm. We used a set of 10K manually created word pronunciations to perform the comparison. We compared the two approaches under two operating conditions. In the first condition, we used 1K manually created word pronunciations for training and 9K for testing. In the second, we divided the 10K words equally into two sets of 5K each. Table 2 shows the percentage of words where the predicted pronunciations were identical to the corresponding reference, i.e. the manual pronunciation. From Table 2, we conclude that our updates to the T2P tool outperform the standard tool by a factor of 2 to 3 in prediction accuracy. On analysis of the pronunciation errors from the modified T2P tool, we found that most of the errors are single phone variations in the phonetic string. Therefore, we adopted the improved approach for subsequent experiments that rely on creating automatic phonetic pronunciations.

Train			Test		
#Wds	T2P	Modified T2P	#Wds	T2P	Modified T2P
1K	36%	98%	9K	12%	22%
5K	29%	96%	5K	14%	42%

Table 5. Percentage of words where the predicted pronunciation from the two different text-to-phone are identical to the reference pronunciation

Next, to perform a systematic comparison of different strategies for creation of pronunciation lexicons, we explored three different scenarios by varying the amount of words with manually created phonetic spellings:

1. *Low-resource* that simulates having pronunciation for only the top-1K most frequent words in the training data.}
2. *Medium-resource* with the top-5K words having manual pronunciations.}
3. *Full-resource* where every word has a manual pronunciation.}

For each of the aforementioned scenarios, we trained the following systems:

1. **P**: Phoneme-based systems that are estimated from the corresponding fraction of the audio transcripts for which every word has a manually created pronunciation.}
2. **G**: Single grapheme-based system trained over the entire training set.}
3. **P+G**: Hybrid phoneme/grapheme approach where the recognition dictionary uses two pronunciations (phonetic and graphemic). For words that have a manual pronunciation we use the phonetic representation and for words that do not have a manual pronunciation we use the grapheme representation. During training, we estimate two sets of context-dependent HMMs. The first set uses phonetic representation and is trained from the corresponding fraction of the audio transcripts for which every word has a manually created pronunciation. The second set uses grapheme representation and is trained over the entire available training set. Thus, the grapheme-based HMMs for all three training scenarios are estimated from the same amount of data. This ensures that the grapheme HMMs use all available training data.}
4. **P+T2P**: In this approach, there is a common set of HMMs that use only phonetic representation. For the words that do not have manual pronunciations, the trained letter-to-sound rules from the modified T2P tool are used to create pronunciations automatically. Therefore, the HMMs are trained over the entire training set. The only difference between the three P+T2P systems is the fraction of words with manual and automatic pronunciations.

Table 2 compares the performance of the systems trained from various dictionary configurations as evaluated on the test set in terms of the word error rate (WER). All results are reported with unsupervised constrained maximum likelihood linear regression (CMLLR) speaker adaptation (Gales, 1998). Decoding was performed with the same 10K vocabulary, except for the system P, where the vocabulary size is restricted to the number of words with manual pronunciations. The out-of-vocabulary (OOV) rate for the test set with the 10K vocabulary is 4%, whereas for system P the OOV rate is 5% for the 5K dictionary and 12% for the 1K dictionary.

System	# of Words with Manual Pronunciation			
	0K	1K	5K	10K (all)
P	-	53.2%	46.2%	45.2%
P + T2P	-	45.7%	45.3%	45.2%
P + G	47.3% (G)	46.8%	45.5%	45.1%

Table 6. WER of systems trained from various dictionary configurations as evaluated on the test set. Decoding was based on the same 10K vocabulary, except System P, where the vocabulary is restricted to the number of words with manual pronunciations.

As one would expect, the grapheme system (System G in parentheses in the P+G row of Table 2 results in the worst performance (WER of 47.3%) compared to the systems with the same vocabulary. On the other hand, the phoneme system (System P), which uses manual pronunciations for every word results in a WER of 45.2% - a 2.1% absolute reduction in WER than the grapheme system.

For the low (1K) and medium (5K) resource scenarios the P+T2P and P+G systems yields better performance than the phoneme system. In particular, the P+T2P system significantly outperforms the P+G system for the low-resource scenario (WER 45.7% vs. 46.8%). For the medium-resource scenario, both P+T2P and P+G systems result in comparable performance. Note that the P+G system uses pronunciation probabilities to assign a different weight to the grapheme and phoneme pronunciations.

5. Machine translation

BBN's Statistical Machine Translation (SMT) engine is a phrase-based translation system based on (Koehn, 2004). Word alignments between source-target sentence pairs are generated using GIZA++ (Och and Ney, 2003). In order to improve the quality of the alignments, word alignments in the forward and backward direction are merged as in (Koehn et al., 2003). Phrase pairs are automatically extracted from the word alignments by merging neighboring alignment groups using a set of rules. The decoder uses a log-linear model of different features to choose between competing translation hypotheses. The parameters of the model are estimated using statistics of the phrase pairs extracted from the word alignments. The interpolation weights are optimized by minimizing the translation errors on a held out development set.

The system uses a variety of techniques for increasing accuracy. Among these is the use of multiple alignments, generated from morphological segmentation, as well as a technique for inducing collocations on the English side of the parallel corpus. This technique uses the Minimum Description Length (MDL) principle to find N-grams whose reduction to a single token reduces the overall number of "bits" needed to encode the document. This has the effect of partially "inflecting" the English, so that it better matches an inflected language on the other side of the corpus. Other recent improvements have been the use of phrase alignment confidence (PAC) (Ananthakrishnan et al., 2009) to deal with data sparseness, and context-dependent lexical smoothing for incorporating context.

In this section, we present several enhancements for statistical machine translation in context of speech-to-speech translation. We illustrate these improvements on Pashto/English MT. We first describe our baseline system for Pashto/English translation.

5.1 Baseline Pashto/English MT

Pashto is an inflected language that follows a Subject Object Verb (SOV) word order versus the Subject Verb Object (SVO) word order of English. Nouns and adjectives in Pashto are inflected for gender, number, and case. Verbs in Pashto are complex both in form and in use. Verbs agree in person and number with either the subject or the object of the sentence depending on the tense and the particular construction. One or more affixes can be attached to a word or to each other to form compound words, and components of compound words can be joined or separated depending on style. The different dialects of Pashto show many non-standard grammatical features, some of which are archaisms or descendants of old forms that are discarded by the literary language.

Translation Direction	#Pashto Words		#English Words	
	Total	Unique	Total	Unique
Pashto-to-English (76K sent. Pairs)	1.3M	20.0K	1.1M	10K
English-to-Pashto (34K sent. Pairs)	520K	13.5K	460K	6.7K

Table 7. Description of Pashto/English parallel data available for SMT training.

The data available for training our SMT engine on Pashto/English is shown in Table 3. The amount of data is significantly smaller than the typical broadcast news translation task, where corpora are of the order of several million sentence pairs. Given the fairly small size of the training data, we trained the translation systems on data from both translation directions. The tuning set (held-out from training) comprises of 2K Pashto sentences and 2.3K English sentences. For validation purposes, we report results on a set of 547 sentences for Pashto-to-English (P2E) and 564 sentences for English-to-Pashto (E2P) with 4 references each.

For translating from Pashto-English, we also segment words in Pashto into its constituent "morphemes", that is prefix, stem, and suffix before training in order to improve the quality of the phrase alignments and subsequently the translation. We used the same decomposition algorithm as in (Riesa et al., 2006) to segment our training data. We manually selected 86 prefixes and 68 suffixes in Pashto. Given the list of predefined affixes and uninflected words we iteratively stripped affixes from the word until a valid combination of affixes and stem was found in a large dictionary. Segmentation into morphemes resulted in a 27% reduction in the size of the Pashto vocabulary. It also reduced the number of unknown tokens (untranslated words) by 38% on the validation set.

The training and decoding was performed as follows. We used GIZA++ (Och and Ney, 2003) to generate the word alignments in the source-target and target-source directions according to IBM Model 4. The merged word alignments are used to generate a phrase translation table which contains source-target phrase pairs and associated statistics. The log-linear model includes features computed from the phrase table as well as the target side language model. We use a 4-gram language model trained on 3M Pashto words for E2P and a 5-gram language model trained on 20M English words for P2E. In our experiments, we optimize the feature weights for maximum BLEU on the held-out tuning set. We then decode the validation set with the same configuration but with the tuned weights instead.

5.2 Phrase alignment confidence

In phrase-based statistical machine translation systems, translation performance is contingent on accurate estimation of the translation model parameters derived from the phrase pair statistics. However, data sparsity, an inherent problem in SMT even with large training corpora, often has an adverse impact on the reliability of the extracted phrase translation pairs. A significant proportion of phrase pairs occurs just once (singletons) or a few times in the training data, often resulting in unreliable estimates of the associated statistics. For instance, the unsmoothed estimate of the translation probability of a singleton phrase pair might be very large, but this estimate could be entirely invalid if the pair originated from a word alignment error. Thus, it is desirable to have a measure of phrase pair quality based on the reliability of the underlying word alignments. The lexical smoothing probability used as a feature in the log-linear decoding framework is a well-known, existing measure of phrase pair reliability. In Ananthakrishnan et al. (2009), the notion of alignment entropy as a measure of automatic word alignment quality was used to estimate a probability distribution over the alignments of a given source word, and thus evaluate the uncertainty (entropy) of its Viterbi alignment in the original parallel corpus. Their experiments indicated that alignment entropy is well-correlated with traditional measures of alignment quality, such as Alignment Error Rate (AER). As an extension of alignment entropy, we introduce a feature called phrase alignment confidence as a measure of phrase pair quality derived from an ensemble of parallel corpora obtained by resampling the original training.

We identify occurrences of the same sentence pair in multiple parallel corpora, and determine, based on the corresponding word alignments, whether the phrase pairs extracted from this sentence pair are consistent across the corpora in which it occurs. The technique of bootstrap resampling (Effron, 1979) can be used to construct such corpora. Assuming the parallel training corpus (the pivot) contains N sentence pairs, we create K independent resamples, each of size N, by sampling the original corpus with replacement. On average, about 63% of sentence pairs in each resample will be unique, the remaining being repetitions. Thus, a given sentence pair in the original corpus can be expected to occur 63 of 100 resamples. We invoke the Expectation-Maximization (EM) algorithm to perform automatic word alignment (based on IBM Model 4) on each of the (K + 1) parallel corpora (pivot + K resamples). As each resample contains a different set of sentence pairs drawn from the pivot, the word alignments in each set can potentially be different. During the phrase extraction process, we scan the pivot and identify valid phrase pairs based on the word alignments. When extracting phrase translations from a given pivot sentence pair (S_i,T_i), we identify all resamples R_i, in which that sentence pair occurs, and determine whether the phrase pairs identified in the pivot sentence pair are consistently valid across the resamples. We define the alignment confidence of a single instance of a phrase pair in the pivot as the ratio of the number of resamples in which that instance is identified as a valid phrase pair to the number of resamples in which the containing sentence pair occurs. Note that this measure is computed for each instance of every phrase pair. For non-singleton phrase pairs, we simply take the average of the phrase alignment confidences of each instance across the pivot corpus. Thus, every phrase pair in the pivot corpus now has an associated confidence score in addition to the original statistics. We refer to this measure as phrase alignment confidence.

The discriminative translation framework of the decoder makes it relatively straightforward to add new features to the system. In order to integrate the phrase alignment confidence feature, we simply add to the log linear model an additional term consisting of the new feature and its corresponding weight.

Tables 8 and 9 present results for P2E and E2P SMT after inclusion of the phrase alignment confidence feature in decoding. We resampled the training corpus (pivot) with replacement to generate K=99 resamples for a total of 100 parallel corpora. We then perform augmented phrase pair extraction where, for each instance of every phrase pair in the pivot corpus, we evaluated its consistency across all resamples in which the containing sentence pair occurs. The augmented phrase table encodes this phrase alignment confidence feature in addition to the original statistics. Integrating the proposed phrase alignment confidence feature improved the BLEU score by 3.5% relative on the P2E validation set and 0.4% relative on the E2P set. We believe that while the proposed feature is useful in its own right, it possesses less discriminative power than the standard lexical smoothing feature. The length of a phrase pair does not play a major role in evaluating the phrase alignment confidence feature, whereas longer pairs are almost always de-emphasized by lexical smoothing. In the future, we plan to extend our work on phrase pair quality measurement by taking phrase pair length and the consistency of within-phrase alignments across the resamples into account, making it more competitive with lexical smoothing as well as giving better additive improvements in combination with the latter. We also plan to evaluate the relative usefulness of phrase alignment confidence with respect to the amount of training data available, and to determine whether its importance increases as the training corpus shrinks in size.

Configuration	BLEU	# of Untrans. Words
Baseline	34.8	80
+ Phrase Alignment Confidence	36.0	80
+ Context-Dependent Lexical Smoothing	36.2	80
+ Back-off to a Bilingual Lexicon	36.2	63

Table 8. Experimental Results for Pashto-to-English Text-to-Text Translation.

Configuration	BLEU	# of Untrans. Words
Baseline	24.8	31
+ Phrase Alignment Confidence	24.9	31
+ Context-Dependent Lexical Smoothing	25.1	31
+ Back-off to a Bilingual Lexicon	25.1	16

Table 9. Experimental Results for English-to-Pashto Text-to-Text Translation.

5.3 Context-dependent lexical smoothing

In our phrase-based decoder, the likelihood of translation from a source phrase S = s_1, s_2, ... s_n to a target phrase T = t_1, t_2, ... t_m is primarily modeled with the *rule translation probability* maximum likelihood estimates:

$$P(S|T) = \frac{N(S,T)}{\sum_{S'} N(S',T)}$$

$$P(T|S) = \frac{N(S,T)}{\sum_{T'} N(S,T')}$$

where N(S,T) is the number of times the rule S→T was extracted from the training corpus. However, translation probability is also modeled with an another feature, known as *lexical smoothing* (Koehn, 2004). The forward lexical smoothing score for the rule S→T is defined as:

$$\prod_{i=1}^{m} \sum_{s \in A(t_i|S,T)} \frac{P(t_i|s)}{\|A(t_i|S,T)\|}$$

where $P(t|s) = N(s,t)/\sum_{t'} N(s,t')$ is the probability of the *word-to-word* translation S→T, and *A(t/S,T)* is the set of source words aligned to t in the rule S→T. In this case, N(t,s) counts the number of times s is aligned to t in the GIZA aligned training data. Also note that either s or t can be NULL.

The backwards lexical smoothing score is analogously:

$$\prod_{i=1}^{m} \sum_{t \in A(s_i|S,T)} \frac{P(s_i|t)}{\|A(s_i|S,T)\|}$$

Note that the lexical smoothing score is computed at the word level without factoring in local context, even though intuitively we know that context is important for both human and machine translation accuracy. On the other hand, the average word-to-word translation

will be seen far more times in the training than the average phrase-to-phrase translation, so the word-level maximum likelihood estimates P(t|s) and P(s|t) will be estimated than the phrase-level maximum likelihood estimates P(S|T) and P(T|S).
Ideally, we would like to harness the increased contextualization of the rule translation probabilities without sacrificing the accuracy of the word-to-word maximum likelihood estimates. To that end, in the *context-dependent lexical smoothing* approach we condition the word translation probabilities on local context, and then interpolate them context-independent probabilities to ensure that the final probabilities are well-estimated.
We currently use *previous word* and *next word* as context types. Formally, these context-dependent lexical probabilities are represented as:

$$P(t|s_i, s_{i-1}) = \frac{N(s_i, s_{i-1}, t)}{\sum_{t'} N(s, s_{i-1}, t')}$$

$$P(t|s_i, s_{i+1}) = \frac{N(s_i, s_{i+1}, t)}{\sum_{t'} N(s, s_{i+1}, t')}$$

Rather than directly interpolating the probabilities or using an explicit back-off model, we simply interpolate the lexical counts:

$$P(t|s_i) = \frac{N(s_i, t) + \alpha N(s, s_{i-1}, t) + \beta N(s, s_{i+1}, t)}{\sum_{t'} N(s_i, t') + \sum_{t'} \alpha N(s, s_{i-1}, t') + \sum_{t'} \beta N(s, s_{i+1}, t')}$$

where $C(s_i)$ is the local context of source word s_i, and the interpolation weights α and β are globally optimized on a tuning set. This type of count-based interpolation acts as an implicit ``back-off'' model, since the more times a particular context type has been seen, the more mass it adds to the final probability.
The interpolated probability P(t|s,C(s)) is used in the standard lexical smoothing formula and this score is used as an additional log-linear decoding feature. We also use context-dependent lexical smoothing in the backwards direction, conditioning on target context.
Tables 8 and 9 summarize the impact of using context-dependent lexical smoothing for P2E and E2P SMT. As shown in the two tables, there is a modest improvement in BLEU scores in both directions.

5.4 Effective use of bilingual lexicon

Often, the heuristics used to determine valid phrases in the phrase extraction step result in unaligned source-target words occurring in the corpora being omitted from the phrase translation table. Hence, a word that appears in the training corpus is not guaranteed to have a translation during decoding. The use of a supplementary bilingual translation lexicon that covers such words improves the coverage of the system. Traditionally bilingual translation lexicons are used as additional training data for machine translation systems and allowed to drive the word alignments. However, the entries in a lexicon have such high phrase translation and lexical probabilities that they can cause serious word sense errors if the particular source word occurs in a different context. If a word that occurs in the lexicon is identified in the input sentence, its corresponding single word translation from the lexicon will almost always be preferred over a longer phrase pair whose source phrase contains that word. We tackle this issue by backing off to entries in

the lexicon only if the source word cannot be translated as part of a source phrase existing in the phrase translation table.
Using a bilingual expert, we created a bilingual lexicon consisting of a total of 30K entries. In our experiments, using the bilingual translation lexicon did not improve the BLEU metric, however it resulted in a 50% reduction in untranslated words for E2P and 21% reduction for P2E as shown in Tables 8 and 9.

6. Evaluation

From 2006 to 2010, BBN TransTalk has been evaluated in several US Goverment sponsored evaluations conduced by an independent third party such as NIST and MITRE. In these evaluations participating S2S systems are evaluated on several dimensions include rate of concept transfer in live interactions with role players, odds of concept transfer on offline data, and automated metrics such as word error rate (WER) and BLEU scores computed on offline recorded audio. User surveys based on questionnaires are also used to measure the ease of use, efficacy of interaction, etc. based on users' impression of the live interaction.
Table 10 summarizes BBN's performance as measured against the following official program metrics in the program for the past three years. The evaluations were typically on a different language and often on different platforms.
High-level Concept Transfer (HCT): This metric is computed from live interaction of users with the system in an allotted time interval (typically 20 minutes). A team of bilingual judges compares the output of the TRANSTAC system to what was spoken by the role-playing US military personnel, i.e., subject matter experts (SMEs) and foreign language speaker (FLS). The judges are asked to rate, on an utterance-by-utterance basis, how well the utterance spoken by the human speaker was translated by the system and how many times the speaker attempted the utterance. When multiple attempts were made, only the best translation was scored. Both English to foreign language and foreign language to English directions were scored. The translation quality has four possible scorings:

1. Unknown – The utterance in the scenario was not attempted by the SME or FLS. A score of "0" is assigned to this category.
2. Inadequate – None of the concepts came across in the utterances.A score of "0" is assigned to this category.
3. Partially adequate – Some of the concepts came across in the utterance.
4. Adequate – All of the concepts came across in the utterance.

Partially adequate are given a score of 0.5, and adequate are given a score of 1. In the case where multiple concepts were provided by the FLE in response to the SME's question, each answer is counted separately. These scores are then aggregated over the entire session, and the transfer rate per ten minutes of conversation is computed.
Odds of Successful Low-Level Concept Transfer (LLCT): This metric is computed using the system output and reference translations on an offline, pre-recorded data set. A bilingual human annotator identifies low-level concepts (such as "car", "door", "black color", etc.) that are correct or incorrectly transferred in the system output. Next, the odds of successful transfer of these low-level concepts are computed by dividing the number of successes by the number of errors. The higher the odds of success, the better the system.

Subject Matter Expert (SME) Utility Assessment (SUA): This metric is computed from responses to questionnaire by SMEs after interacting with the system in any given session. A utility score is computed by aggregating scores across sessions for each type of question. The questions in the SME questionnaire range from: "I found the system easy to understand in this interaction" to "I would use this system in the field in its current state of functionality".

Table 10 describes the performance of BBN systems in the evaluations in reverse chronological order with most recent evaluations at the top of the table.

Eval. Date	Language	Platform	HCT		LLCT		SUA
			E2F	F2E	E2F	F2E	
Aug 2010	Dari	Smartphone	15 (1st)	25 (1st)	3.3 (2nd)	1.5 (1st)	1st
April 2010	Pashto	Smartphone	19 (1st)	30 (1st)	4.2 (1st)	3.0 (1st)	1st
June 2009	Dari	UMPC	13 (1st)	14 (1st)	3.5 (1st)	1.6 (1st)	1st
Nov 2008	Iraqi	Laptop	22 (2nd)	28 (2nd)	7.3 (1st)	5.6 (1st)	1st

Table 10. Performance of BBN TransTalk in recent DARPA TRANSTAC evaluations.

7. Conclusions

We have presented our speech-to-speech translation system, TransTalk, and outlined several techniques for overcoming challenges in languages that it has been configured in. For ASR, we described an approach for configuring the ASR system with limited amount of manual pronunciations. Our approach extends existing approaches for languages that have significant mismatch in number of phonemes and graphemes, and shows that comparable performance to a full lexicon can be achieved by creating manual pronunciations for a small fraction of words in the vocabulary. For MT, we discussed techniques for overcoming challenges due to data sparsity such as the use of phrase alignment confidence and effective backoff to a bilingual dictionary. We also presented a method for evaluating the effectiveness of different user confirmation strategies, and shown that back-translation provides higher precision than the simple strategy of reading back the ASR, at the expense of recall.

8. References

Akbacak, M., Franco, H., Frandsen, M., Hasan, S., Jameel, H., Kathol, A., Khadivi, S., Lei, X., Mandal, A., Mansour, S., Precoda, K., Richey, C., Vergyri, D., Wang, W., Yang, M., Zheng, J., 2009. Recent advances in sri's iraqcomm: Iraqi arabic-english speech-to-speech translation system. In: ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing.

Ananthakrishnan, S., Prasad, R., Natarajan, P., 2009.Alignment entropy as an automated measure of bitext fidelity for statistical machine translation.In: Proceedings of the 7th International Conference on Natural Language Processing.

Belvin, R., Ettelaie, E., Gandhe, S., Georgiou, P., Knight, K., Marcu, D., Narayanan, S., Traum, D., 2005. Transonics: A practical speech-to-speech translator for english-farsi medical dialogues. In: Proceedings of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING). pp. 89-92.

Bikel, D. M., Schwartz, R., Weischedel, R. M., 1999.An algorithm that learns what's in a name.In: Machine Learning Special Issue on Natural Language Learning.

Billa, J., Noamany, M., Srivastava, A., Liu, D., Stone, R., Xu, J., Makhoul, J., Kubala, F., 2002. Audio indexing of arabic broadcast news. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing.Vol. 1, pp. I-5 – I-8.

Black, A., Lenzo, K., Pagel, V., 1998. Issues in building general letter to sound rules. In: ESCA Workshop on Speech Synthesis. pp. 77-80.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., Ue±ng, N., 2004.Con¯dence estimation for machine translation.In: COLING '04: Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, Morristown, NJ, USA, p. 315.

Efron, B., 1979. Bootstrap methods: Another look at the jackknife. The Annals of Statistics 7 (1), pp. 1-26.

Gales, M., 1998.Maximum likelihood linear transformations for hmm-based speech recognition.Computer Speech and Language 12 (2), 75-98.

Gao, Y., Gu, L., Zhou, B., Sarikaya, R., Afify, M., Kuo, H.-k., Zhu, W.-z., Deng, Y., Prosser, C., Zhang, W., Besacier, L., 2006. Ibm mastor system: Multilingual automatic speech-to-speech translator. In: Proceedings of HLT Medical Speech Translation Workshop.

Koehn, P., 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In: AMTA. pp. 115-124.

Koehn, P., Och, F. J., Marcu, D., 2003.Statistical phrase-based translation. In: NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Association for Computational Linguistics, Morristown, NJ, USA, pp. 48-54.

Magimai-Doss, M., Bengio, S., Bourlard, H., may. 2004. Joint decoding for phoneme-grapheme continuous speech recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 1.

McCallum, A., Nigam, K., 1998. A comparison of event models for naïve bayes text classi¯cation. In: Proceedings of AAAI-98 Workshop on Learning for Text Categorization. AAAI Press, pp. 41-48.

Nguyen, L., Schwartz, R., 1997.Efficient 2-pass n-best decoder. In: DARPA Speech Recognition Workshop. pp. 167-170.

Och, F. J., Ney, H., March 2003.A systematic comparison of various statistical alignment models. Computational Linguistics 29 (1), 19-51.

Prasad, R., Moran, C., Choi, F., Meermeier, R., Saleem, S., C., K., Stallard,D., Natarajan, P., 2008. Name aware speech-to-speech translation for english/iraqi. pp. 249-252.

Prasad, R., Tsakalidis, S., Bulyko, I., Kao, C.-l., Natarajan, P., 2010.Pashto speech recognition with limited pronunciation lexicon. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. pp. 5086-5089.

Riesa, J., Mohit, B., Knight, K., Marcu, D., 2006. Building an english-iraqi arabic machine translation system for spoken utterances with limited resources. In: Proceedings of ISCA Interspeech.

Stallard, D., Choi, F., Kao, C.-L., Krstovski, K., Natarajan, P., Prasad, R., Saleem, S., Subramanian, K., 2007.The bbn 2007 displayless English/Iraqi speech-to-speech translation system.In: Proceedings of ISCA INTERSPEECH. ISCA, pp. 2817-2820.

Speech and Language Technologies
Edited by Prof. Ivo Ipsic

ISBN 978-953-307-322-4
Hard cover, 344 pages
Publisher InTech
Published online 21, June, 2011
Published in print edition June, 2011

This book addresses state-of-the-art systems and achievements in various topics in the research field of speech and language technologies. Book chapters are organized in different sections covering diverse problems, which have to be solved in speech recognition and language understanding systems. In the first section machine translation systems based on large parallel corpora using rule-based and statistical-based translation methods are presented. The third chapter presents work on real time two way speech-to-speech translation systems. In the second section two papers explore the use of speech technologies in language learning. The third section presents a work on language modeling used for speech recognition. The chapters in section Text-to-speech systems and emotional speech describe corpus-based speech synthesis and highlight the importance of speech prosody in speech recognition. In the fifth section the problem of speaker diarization is addressed. The last section presents various topics in speech technology applications like audio-visual speech recognition and lip reading systems.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

David Stallard, Rohit Prasad, Prem Natarajan, Fred Choi, Shirin Saleem, Ralf Meermeier, Kriste Krstovski, Shankar Ananthakrishnan and Jacob Devlin (2011). The BBN TransTalk Speech-to-Speech Translation System, Speech and Language Technologies, Prof. Ivo Ipsic (Ed.), ISBN: 978-953-307-322-4, InTech, Available from: http://www.intechopen.com/books/speech-and-language-technologies/the-bbn-transtalk-speech-to-speech-translation-system

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen